Causal Representation Learning for Out-of-Distribution Recommendation

Wenjie Wang[1], Xinyu Lin[1], Fuli Feng[2*], Xiangnan He[2], Min Lin[3], and Tat-Seng Chua[1]
[1]National University of Singapore, [2]University of Science and Technology of China, [3]Sea AI Lab
{wenjiewang96,xylin1028,fulifeng93,xiangnanhe}@gmail.com,linmin@sea.com,dcscts@nus.edu.sg

ABSTRACT

Modern recommender systems learn user representations from historical interactions, which suffer from the problem of user feature shifts, such as an income increase. Historical interactions will inject out-of-date information into the representation in conflict with the latest user feature, leading to improper recommendations. In this work, we consider the Out-Of-Distribution (OOD) recommendation problem in an OOD environment with user feature shifts. To pursue high fidelity, we set additional objectives for representation learning as: 1) strong OOD generalization and 2) fast OOD adaptation.

This work formulates and solves the problem from a causal view. We formulate the user feature shift as an *intervention* and OOD recommendation as *post-intervention inference* of the interaction probability. Towards the learning objectives, we embrace causal modeling of the generation procedure from user features to interactions. However, the unobserved user features cannot be ignored, which make the estimation of the interaction probability intractable. We thus devise a new Variational Auto-Encoder for causal modeling by incorporating an encoder to infer unobserved user features from historical interactions. We further perform *counterfactual inference* to mitigate the effect of out-of-date interactions. Moreover, a decoder is used to model the interaction generation procedure and perform post-intervention inference. Fast adaptation is inherent owing to the reuse of partial user representations. Lastly, we devise an extension to encode fine-grained causal relationships from user features to preference. Empirical results on three datasets validate the strong OOD generalization and fast adaptation abilities of the proposed method.

CCS CONCEPTS

• **Information systems → Personalization**; **Recommender systems**.

KEYWORDS

Causal Representation Learning, OOD Recommendation, User Feature Shifts

ACM Reference Format:
Wenjie Wang, Xinyu Lin, Fuli Feng, Xiangnan He, Min Lin, and Tat-Seng Chua. 2022. Causal Representation Learning for Out-of-Distribution Recommendation. In *Proceedings of the ACM Web Conference 2022 (WWW '22), April 25–29, 2022, Virtual Event, Lyon, France.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3485447.3512251

1 INTRODUCTION

Recommender systems have been widely deployed for personalized information filtering to alleviate information explosion on the Web [14, 27]. As the core of recommender models, learning representation of user preference relies on historical interactions. Existing approaches are mainly based on the Independent and Identically Distributed (IID) assumption of the interactions between training and testing periods. However, user feature shifts (*e.g.,* an income increase) are common in practice, which will affect the user preference and behaviors. As such, the representations learned with out-of-date interactions (*e.g.,* purchases of cheap copies) will cause improper recommendations (*cf.* Figure 1). We reveal that existing recommender models encounter significant performance drop in an OOD environment with user feature shifts (*cf.* Table 2), thus hurting user experience and leading to notorious issues like the unfairness across users. As such, it is essential to consider the OOD recommendation problem.

OOD recommendation has received little scrutiny. Existing approaches that have the potential to deal with user feature shifts mainly fall into three categories. 1) Feature-based models [27], which can be adapted to the OOD environment by model inference with the latest user features. However, they still suffer from the out-of-date interactions since they are unable to disentangle the effects of user features and historical interactions. 2) Disentangled recommendation [22] aims to learn factorized representations for user preference, which can be more robust to distribution shifts. Nevertheless, previous studies mostly ignore user features and encode the out-of-date interactions in the representation [41]. 3) Model re-training [28, 36] also facilitates adaptation, but faces a dilemma between the re-training frequency and computation cost. Moreover, it requires to collect new interactions after the feature shifts, which means that inappropriate items are still recommended until sufficient new interactions are collected.

To strengthen recommender systems, we require the representation learning of user preference to pursue two objectives: 1) strong OOD generalization; and 2) fast adaptation. OOD generalization means that the model can infer accurate user preference for the latest user features, *i.e.,* directly adapting to the OOD environment. Once very few new interactions are collected from the OOD environment, fast adaptation implies that the model can be quickly and accurately updated. To achieve the two learning objectives, the key lies in the abilities to: 1) figure out the mechanism to understand

[*] Corresponding author: Fuli Feng (fulifeng93@gmail.com). This research is supported by the Sea-NExT Joint Lab, and the National Natural Science Foundation of China (62121002).

WWW '22, April 25–29, 2022, Virtual Event, Lyon, France
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9096-5/22/04…$15.00
https://doi.org/10.1145/3485447.3512251

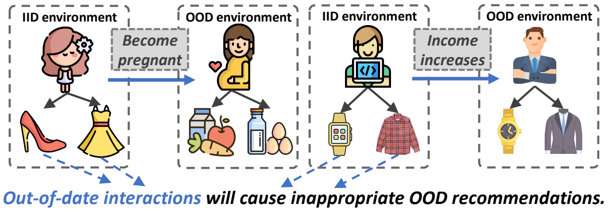


Figure 1: Examples of OOD recommendation.

how feature shifts affect user preference; 2) mitigate the effect of out-of-date interactions on OOD recommendation; and 3) reuse partial unchanged user representations to accelerate adaptation [32].

We resort to causal language and scrutinize the cause-effect factors in the interaction generation procedure, which are abstracted as a causal graph in Figure 2. The causal graph describes the causal relationships from user features (E_1 and E_2) to user preference (Z_1 and Z_2), and user interactions (D). Note that we split the user features into the observed group (E_1) and unobserved group (E_2), and set two types of preference depending on whether it is affected by the observed features (Z_1) or not (Z_2). Existing methods construct the preference representation by encoding both E_1 and D, and thus suffer from the out-of-date interactions in the OOD environment. From the causal view, OOD recommendation is indeed the *post-intervention inference* of interaction probabilities $P(D|do(E_1 = \boldsymbol{e}_1'), E_2)$, where the feature shift from $E_1 = \boldsymbol{e}_1$ to $E_1 = \boldsymbol{e}_1'$ is formulated as an intervention [25]. Furthermore, as $do(E_1 = \boldsymbol{e}_1')$ only affects Z_1, we can facilitate fast adaptation by reusing the unaffected part Z_2 [25, 32].

Towards this end, we propose a Causal OOD Recommendation (COR) framework that models the interaction generation procedure according to the causal graph. The challenge of this framework is to deal with the unobserved features E_2, which makes the estimation of $P(D|do(E_1 = \boldsymbol{e}_1'), E_2)$ intractable. To solve this challenge, we resort to variational inference and design a new Variational Auto-Encoder (VAE) with an encoder to infer the unobserved E_2 from the historical interactions D and observed E_1 by modeling $P(E_2|D, E_1)$. Besides, a decoder network is needed to estimate $P(D|E_1, E_2)$. Once learned, we can perform post-intervention inference by feeding the latest user features $\boldsymbol{e}_1'$ to the VAE. Moreover, to prevent potential impacts from out-of-date interactions in D, we adopt a *counterfactual inference* to block the harmful effect of D. As to fast adaptation with new interactions, we reuse Z_2 and only update Z_1 via fine-tuning. Furthermore, we design an extension to demonstrate that COR is able to capture more fine-grained causal relationships between E_1, E_2, and Z_1. Extensive experiments on a synthetic dataset and two real-world ones validate that COR is able to achieve strong OOD generalization and fast adaptation with comparable IID performance. We release the code and data at https://github.com/Linxyhaha/COR.

To summarize, the main contributions of this work are as follows:

- We study a new OOD recommendation problem, formulating and solving it from a causal view.
- We propose a Causal OOD Recommendation framework, which performs causal modeling and inference to handle feature shifts.
- Extensive experiments on three datasets demonstrate the superiority of COR on enhancing OOD generalization and fast adaptation while maintaining the IID performance.

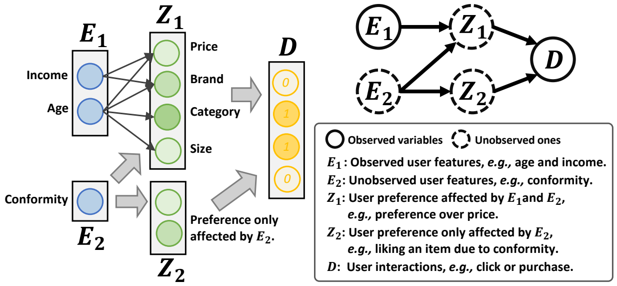

E_1
Income
Age
Z_1
Price
Brand
Category
Size
Conformity
E_2
Z_2
Preference only affected by E_2.
D
0
1
1
0
E_1
Z_1
E_2
Z_2
D
Observed variables
Unobserved ones
E_1: Observed user features, e.g., age and income.
E_2: Unobserved user features, e.g., conformity.
Z_1: User preference affected by $E_1$and E_2, e.g., preference over price.
Z_2: User preference only affected by E_2, e.g., liking an item due to conformity.
D: User interactions, e.g., click or purchase.

Figure 2: Causal graph of the interaction generation process.

2 RECOMMENDATION RE-FORMULATION

In this section, we inspect the interaction generation process and formulate OOD recommendation from a causal view.

Causal View of User Interaction Generation In Figure 2, we abstract the interaction generation process as a causal graph. We explain its rationality as follows.

- E_1, E_2 represent observed user features (*e.g.*, age and income) and unobserved user features (*e.g.*, conformity and social networks), respectively. Most recommender systems access partial user features due to privacy restriction and device limitation.
- Z_1, Z_2 represent the latent user preference, which is split into two groups regarding whether it is affected by E_1. Z_2 is separated because there always exists user preference unaffected by E_1.
- D denotes user's interaction status over items.
- $(E_1, E_2) \rightarrow Z_1$ and $E_2 \rightarrow Z_2$ denote that user preference is determined by user features. For instance, *income* affects the preference over *price* and *brand*.
- $(Z_1, Z_2) \rightarrow D$ means that user's interaction status over items is determined by user preference.

Formulation of OOD Recommendation We use $u \in \{1, ..., U\}$ and $i \in \{1, ..., I\}$ to index users and items, respectively. For a user u, the recommender models aim to learn the user preference representation $[z_1, z_2]$ from the observed features $E_1 = \boldsymbol{e}_1$ and historical interactions $D = \boldsymbol{d} \in \{0, 1\}^I$ which is a multi-hot vector with $d_i = 1$ indicating an interaction between item i and the user[1]. Based on the representation $[z_1, z_2]$, the model then infers the interaction probabilities over items to make recommendations. This work studies an unexplored **OOD recommendation** problem, where the user feature encounters a shift from $\boldsymbol{e}_1$ to $\boldsymbol{e}_1'$, such as an increased income[2]. From a causal view, we term the feature shift as an *intervention* [25], denoted as $do(E_1 = \boldsymbol{e}_1')$. Accordingly, the recommender model should be able to infer the post-intervention distribution of D. To evaluate the OOD recommendation performance, we propose two specific tasks:

1) **OOD generalization**, which evaluates the generalization ability of a model when the intervention $do(E_1 = \boldsymbol{e}_1')$ is known but user interactions after the intervention are unavailable.
2) **Fast adaptation** assumes that very few post-intervention user interactions are collectable from the OOD environment, and evaluates how quickly and accurately the model adapts to the OOD environment.

[1] For notation brevity, we omit the subscript u in $\boldsymbol{e}_1$, $\boldsymbol{d}$, z_1, and z_2.

[2] As an initial attempt, we ignore shifts of unobserved $\boldsymbol{e}_2'$, *e.g.*, mood changes, which are left to future work since the detection of such changes is still an open problem.

3 CAUSAL OOD RECOMMENDATION

In this section, we first detail the causal modeling of the interaction generation process via a VAE. Based on the VAE, we conduct causal inference for OOD generalization and adopt a fine-tuning strategy for fast adaptation. Lastly, we devise an extension to encode the fine-grained causal graph into COR.

3.1 Causal Representation Learning

We consider the interaction generation process presented in Figure 2 to build the recommender models. Specifically, for each user u with observed feature $\boldsymbol{e}_1$ and historical interactions $\boldsymbol{d}$, let a K-dimension latent vector $\boldsymbol{e}_2$ denote the unobserved feature, which is sampled from a standard Gaussian prior [19]. From $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$, we calculate the distribution of the user preference and sample z_1 and z_2 to produce the interaction probability over I items. Inspired by prior studies [19, 48], we assume the user preference and interaction follow factorized Gaussian and multinomial priors, respectively. Formally, $\boldsymbol{e}_2$, z_1, z_2, and $\boldsymbol{d}$ are drawn from:

$$\begin{cases} \boldsymbol{e}_2 \sim \mathcal{N}(0, \mathrm{I}_K), \\ z_1 \sim \mathcal{N}\left(\boldsymbol{\mu}_{\theta_1}(\boldsymbol{e}_1, \boldsymbol{e}_2), \mathrm{diag}\{\boldsymbol{\sigma}^2_{\theta_1}(\boldsymbol{e}_1, \boldsymbol{e}_2)\}\right), \\ z_2 \sim \mathcal{N}\left(\boldsymbol{\mu}_{\theta_2}(\boldsymbol{e}_2), \mathrm{diag}\{\boldsymbol{\sigma}^2_{\theta_2}(\boldsymbol{e}_2)\}\right), \\ \boldsymbol{d} \sim \mathrm{Mult}\left(N, \pi\left(f_{\theta_3}(z_1, z_2)\right)\right). \end{cases} \tag{1}$$

- $(E_1, E_2) \rightarrow Z_1$: $\boldsymbol{\mu}_{\theta_1}(\boldsymbol{e}_1, \boldsymbol{e}_2)$ denotes the *mean* of the Gaussian distribution estimated from $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ by function $f_{\theta_1}(\boldsymbol{e}_1, \boldsymbol{e}_2)$ parameterized by θ_1; $\mathrm{diag}\{\boldsymbol{\sigma}^2_{\theta_1}(\boldsymbol{e}_1, \boldsymbol{e}_2)\}$ denotes the *diagonal covariance*[3] of the Gaussian distribution.
- $E_2 \rightarrow Z_2$: similarly, $f_{\theta_2}(\boldsymbol{e}_2)$ calculates the mean and diagonal covariance for the Gaussian distribution of z_2.
- $(Z_1, Z_2) \rightarrow D$: $\boldsymbol{d}$ is drawn from the multinomial distribution with the parameters of $N = \sum_{i=1}^{I} d_i$ and $\pi\left(f_{\theta_3}(z_1, z_2)\right)$. N is the number of interactions of user u, and $\pi(\cdot)$ denotes the *softmax* function to normalize the output of $f_{\theta_3}(z_1, z_2)$.

A default choice to optimize the model parameters $\theta = \{\theta_1, \theta_2, \theta_3\}$ is through the reconstruction of interaction history $\boldsymbol{d}$ based on the observed feature $\boldsymbol{e}_1$. Specifically, given a user u with $\boldsymbol{e}_1$ and $\boldsymbol{d}$, we maximize the log-likelihood $\log p(\boldsymbol{d}|\boldsymbol{e}_1)$. Formally,

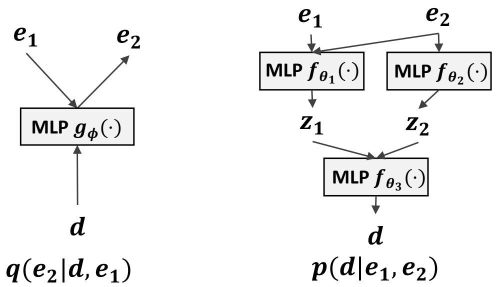
$$\log p(\boldsymbol{d}|\boldsymbol{e}_1) = \log \int p(\boldsymbol{d}, \boldsymbol{e}_2|\boldsymbol{e}_1) d\boldsymbol{e}_2 = \log \int p(\boldsymbol{d}|\boldsymbol{e}_1, \boldsymbol{e}_2) p(\boldsymbol{e}_2) d\boldsymbol{e}_2. \tag{2}$$

Undeniably, Eq. (2) is intractable due to the integral over unobserved features $\boldsymbol{e}_2$ [51]. To tackle this optimization challenge, we resort to variational inference [19], which introduces a variational distribution $q(\boldsymbol{e}_2|\cdot)$ to produce the evidence lower bound (ELBO) of Eq. (2). In particular,

$$\log p(\boldsymbol{d}|\boldsymbol{e}_1) = \log \int p(\boldsymbol{d}|\boldsymbol{e}_1, \boldsymbol{e}_2) p(\boldsymbol{e}_2) \frac{q(\boldsymbol{e}_2|\cdot)}{q(\boldsymbol{e}_2|\cdot)} d\boldsymbol{e}_2 \tag{3a}$$

$$\geq \mathbb{E}_{q(\boldsymbol{e}_2|\cdot)} \left[\log \frac{p(\boldsymbol{d}|\boldsymbol{e}_1, \boldsymbol{e}_2) p(\boldsymbol{e}_2)}{q(\boldsymbol{e}_2|\cdot)} \right] \tag{3b}$$

$$= \mathbb{E}_{q(\boldsymbol{e}_2|\cdot)} \left[\log p(\boldsymbol{d}|\boldsymbol{e}_1, \boldsymbol{e}_2) \right] - \mathrm{KL}(q(\boldsymbol{e}_2|\cdot) \| p(\boldsymbol{e}_2)), \tag{3c}$$


$\boldsymbol{e_1}$
$\boldsymbol{e_2}$
MLP $g_{\boldsymbol{\phi}}(\cdot)$
$\boldsymbol{d}$
$q(\boldsymbol{e_2}|\boldsymbol{d}, \boldsymbol{e_1})$
$\boldsymbol{e_1}$
$\boldsymbol{e_2}$
MLP $f_{\boldsymbol{\theta_1}}(\cdot)$
MLP $f_{\boldsymbol{\theta_2}}(\cdot)$
$\boldsymbol{z_1}$
$\boldsymbol{z_2}$
MLP $f_{\boldsymbol{\theta_3}}(\cdot)$
$\boldsymbol{d}$
$p(\boldsymbol{d}|\boldsymbol{e_1}, \boldsymbol{e_2})$

(a) Encoder Network. **(b) Decoder Network.**

Figure 3: Illustration of the encoder and decoder networks.

where the first term in Eq. (3c) is for reconstruction and the second one regularizes the Kullback-Leibler (KL) divergence between $q(\boldsymbol{e}_2|\cdot)$ and the prior $p(\boldsymbol{e}_2)$. The logic is that maximizing the ELBO in Eq. (3c) will increase the log-likelihood in Eq. (2). Thereafter, to calculate the ELBO, we adopt the encoder and decoder networks to model $q(\boldsymbol{e}_2|\cdot)$ and $p(\boldsymbol{d}|\boldsymbol{e}_1, \boldsymbol{e}_2)$, respectively.

3.1.1 *Encoder Network*

We define $q(\boldsymbol{e}_2|\cdot) = q(\boldsymbol{e}_2|\boldsymbol{d}, \boldsymbol{e}_1)$, which predicts $\boldsymbol{e}_2$ from $\boldsymbol{d}$ and $\boldsymbol{e}_1$ (Figure 3(a)) because the historical interactions and observed user features are likely to indicate the unobserved user features [19]. For example, users' purchase behaviors and age can reflect user conformity. To pursue efficient estimation of $q(\boldsymbol{e}_2|\boldsymbol{d}, \boldsymbol{e}_1)$, we use *amortized inference* [13] and incorporate an encoder network $g_\phi(\cdot)$. Formally,

$$q(\boldsymbol{e}_2|\boldsymbol{d}, \boldsymbol{e}_1) = \mathcal{N}\left(\boldsymbol{e}_2; \boldsymbol{\mu}_\phi(\boldsymbol{d}, \boldsymbol{e}_1), \mathrm{diag}\{\boldsymbol{\sigma}^2_\phi(\boldsymbol{d}, \boldsymbol{e}_1)\}\right), \tag{4}$$

where $\boldsymbol{\mu}_\phi(\cdot)$ and $\boldsymbol{\sigma}_\phi(\cdot)$ are obtained by the encoder $g_\phi(\cdot)$, *i.e.*, $g_\phi(\boldsymbol{d}, \boldsymbol{e}_1) = [\boldsymbol{\mu}_\phi(\boldsymbol{d}, \boldsymbol{e}_1), \boldsymbol{\sigma}_\phi(\boldsymbol{d}, \boldsymbol{e}_1)] \in \mathbb{R}^{2K}$. In this work, we implement $g_\phi(\cdot)$ by a multi-layer perceptron (MLP), where the number of layers and hidden units are hyper-parameters.

3.1.2 *Decoder Network*

As illustrated in Figure 3(b), we factorize $p(\boldsymbol{d}|\boldsymbol{e}_1, \boldsymbol{e}_2)$ according to Eq. (1) and have

$$p(\boldsymbol{d}|\boldsymbol{e}_1, \boldsymbol{e}_2) = \int \int p(z_1|\boldsymbol{e}_1, \boldsymbol{e}_2) p(z_2|\boldsymbol{e}_2) p(\boldsymbol{d}|z_1, z_2) dz_1 dz_2, \tag{5}$$

where the parameters of $p(z_1|\boldsymbol{e}_1, \boldsymbol{e}_2)$ and $p(z_2|\boldsymbol{e}_2)$ are estimated by $f_{\theta_1}(\boldsymbol{e}_1, \boldsymbol{e}_2)$ and $f_{\theta_2}(\boldsymbol{e}_2)$, respectively. In this work, we implement them by two MLP models. Formally, $f_{\theta_1}(\boldsymbol{e}_1, \boldsymbol{e}_2) = [\boldsymbol{\mu}_{\theta_1}(\boldsymbol{e}_1, \boldsymbol{e}_2), \boldsymbol{\sigma}_{\theta_1}(\boldsymbol{e}_1, \boldsymbol{e}_2)]$ and $f_{\theta_2}(\boldsymbol{e}_2) = [\boldsymbol{\mu}_{\theta_2}(\boldsymbol{e}_2), \boldsymbol{\sigma}_{\theta_2}(\boldsymbol{e}_2)]$.

- **Approximation of** $p(\boldsymbol{d}|\boldsymbol{e}_1, \boldsymbol{e}_2)$**.** Even if we obtain the estimations of $p(z_1|\boldsymbol{e}_1, \boldsymbol{e}_2)$ and $p(z_2|\boldsymbol{e}_2)$, the calculation of $p(\boldsymbol{d}|\boldsymbol{e}_1, \boldsymbol{e}_2)$ is challenging due to the costly integral over the latent variables z_1 and z_2. To pursue high efficiency, we resort to Monte Carlo (MC) sampling, which samples z_1 and z_2 to approximate $p(\boldsymbol{d}|\boldsymbol{e}_1, \boldsymbol{e}_2)$[4]. Formally,

$$p(\boldsymbol{d}|\boldsymbol{e}_1, \boldsymbol{e}_2) \approx \frac{1}{L} \frac{1}{M} \sum_{a=1}^{L} \sum_{b=1}^{M} p\left(\boldsymbol{d}|z_1^a, z_2^b\right), \tag{6}$$

where L and M are the sample numbers; z_1^a and z_2^b are drawn from $p(z_1|\boldsymbol{e}_1, \boldsymbol{e}_2)$ and $p(z_2|\boldsymbol{e}_2)$, respectively. Nevertheless, Eq. (6) is still computationally costly due to calculating the conditional

[3] The factors in z_1 are conditionally independent given the parents $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$, which follows the d-separation principle [25] and saves parameters [19, 48]

[4] We use MC sampling instead of variational inference to avoid unnecessary prior hypothesis over the $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ of z_1 and z_2 in Eq. (1) (*e.g.*, $\mathcal{N}(0, 1)$).

Algorithm 1 Inference Pipeline of COR for OOD Generalization

```
Input: latest features $\boldsymbol{e}_1'$ and historical interactions $\boldsymbol{d}$ of user $u$;
    trained networks $g_\phi(\cdot)$, $f_{\theta_1}(\cdot)$, $f_{\theta_2}(\cdot)$, and $f_{\theta_3}(\cdot)$.
  Draw $\boldsymbol{e}_2$ via $g_\phi(\boldsymbol{d}, \boldsymbol{e}_1')$.
  Abduction: draw $\boldsymbol{z}_2$ via $f_{\theta_2}(\boldsymbol{e}_2)$.
  Action: draw $\boldsymbol{e}_2'$ via $g_\phi(\boldsymbol{0}, \boldsymbol{e}_1')$ under $do(D = \boldsymbol{0})$; then draw $\boldsymbol{z}_1'$
    via $f_{\theta_1}(\boldsymbol{e}_1', \boldsymbol{e}_2')$.
  Prediction: calculate $\boldsymbol{d}' = f_{\theta_3}(\boldsymbol{z}_1', \boldsymbol{z}_2)$.
Output: the interaction probability $\boldsymbol{d}'$ for user $u$.
```

probability many times (*i.e.,* $L \times M$). We thus further conduct a widely used approximation [11, 37], which is formulated as:

$$p(\boldsymbol{d}|\boldsymbol{e}_1, \boldsymbol{e}_2) \approx p\left(\boldsymbol{d}\,\middle|\,\frac{1}{L}\sum_{a=1}^{L} \boldsymbol{z}_1^a, \frac{1}{M}\sum_{b=1}^{M} \boldsymbol{z}_2^b\right) = p(\boldsymbol{d}|\bar{\boldsymbol{z}}_1, \bar{\boldsymbol{z}}_2), \tag{7}$$

where the approximation error (*i.e., Jensen gap* [2]) can be well bounded for most functions to calculate $p(\boldsymbol{d}|\bar{\boldsymbol{z}}_1, \bar{\boldsymbol{z}}_2)$ [11].

Thereafter, we can estimate the parameters of $p(\boldsymbol{d}|\bar{\boldsymbol{z}}_1, \bar{\boldsymbol{z}}_2)$ by an MLP $f_{\theta_3}(\cdot)$, which produces an interaction probability distribution $\boldsymbol{d}'$ over I items. Next, the reconstruction term $\log p(\boldsymbol{d}|\boldsymbol{e}_1, \boldsymbol{e}_2)$ in Eq. (3c) is obtained by

$$\log p(\boldsymbol{d}|\bar{\boldsymbol{z}}_1, \bar{\boldsymbol{z}}_2) \overset{c}{=} \sum_{i=1}^{I} d_i \log \pi_i\left(f_{\theta_3}(\bar{\boldsymbol{z}}_1, \bar{\boldsymbol{z}}_2)\right), \tag{8}$$

where d_i denotes whether user u interacts with item i, and $\pi_i\left(f_{\theta_3}(\cdot)\right)$ refers to the prediction score of item i after the softmax normalization $\pi(\cdot)$ over the output of $f_{\theta_3}(\cdot)$. Intuitively, Eq. (8) calculates the reconstruction probability of drawing $\boldsymbol{d}$ from the multinomial distribution by sampling N times, where $N = \sum_{i=1}^{I} d_i$.

3.1.3 *COR Optimization*.

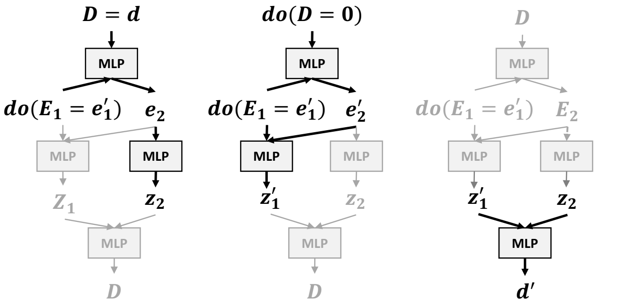
To summarize, we maximize the ELBO in Eq. (3c) to optimize the parameters of COR (*i.e.,* ϕ and $\theta = \{\theta_1, \theta_2, \theta_3\}$) through stochastic gradient descent. To enable the back-propagation of gradients through the sampling operations, we leverage the *reparametrization trick* [17, 19]. In addition, following [19, 22], we also introduce the *KL annealing* [19] with a hyper-parameter β to restrict the regularization of KL divergence. Formally, the ELBO objective for user u is:

$$\mathbb{E}_{q_\phi(\boldsymbol{e}_2|\boldsymbol{d}, \boldsymbol{e}_1)}\left[\log p_\theta(\boldsymbol{d}|\boldsymbol{e}_1, \boldsymbol{e}_2)\right] - \beta \cdot \mathrm{KL}(q_\phi(\boldsymbol{e}_2|\boldsymbol{d}, \boldsymbol{e}_1) \| p(\boldsymbol{e}_2)). \tag{9}$$

During training, the overall objective is calculated by averaging the ELBO over all users. In the inference stage, COR will rank items by $\boldsymbol{d}' = f_{\theta_3}(\bar{\boldsymbol{z}}_1, \bar{\boldsymbol{z}}_2)$ and recommend the top-ranked items.

3.2 Causal Inference for OOD Recommendation

We have introduced the VAE for causal modeling of the interaction generation procedure. Next, we elaborate how to infer the post-intervention interaction probability $p(\boldsymbol{d}'|\boldsymbol{e}_1', \boldsymbol{e}_2)$ (*i.e.,* $P(D|do(E_1 = \boldsymbol{e}_1'), \boldsymbol{e}_2)$). A straightforward way is first feeding $\boldsymbol{d}$ and $\boldsymbol{e}_1'$ into the encoder $g_\phi(\cdot)$ to sample $\boldsymbol{e}_2$; and then feeding $\boldsymbol{e}_1'$ and $\boldsymbol{e}_2$ to the decoder to calculate $\boldsymbol{d}'$.



(a) Abduction. (b) Action. (c) Prediction.

Figure 4: Illustration of counterfactual inference.

• **Counterfactual Inference.** Nevertheless, the straightforward solution faces the risk to bring in the bad effect of $\boldsymbol{d}$ (*i.e.,* impact of out-of-date interactions) when inferring $\boldsymbol{e}_2$ because the encoder takes historical interactions as inputs (*i.e.,* $g_\phi(\boldsymbol{d}, \boldsymbol{e}_1')$). To avoid such out-of-date information in conflict with the changed feature $\boldsymbol{e}_1'$, we cut off the bad effect from $\boldsymbol{e}_2$ to $\boldsymbol{z}_1$, while reserving the good effect from $\boldsymbol{e}_2$ to $\boldsymbol{z}_2$ because $\boldsymbol{z}_2$ is not affected by $do(E_1 = \boldsymbol{e}_1')$ according to the causal graph and should be stable in the OOD environment. Towards the goal, we propose a counterfactual inference strategy, which imagines *what the predicted user interactions D would be if Z_1 were not affected by $\boldsymbol{d}$*. According to the three-step definition of counterfactual inference (*cf.* Theorem 7.1.7 in [25]), we design the inference strategy as:

- **Abduction**: estimate $\boldsymbol{z}_2$ based on the factual $D = \boldsymbol{d}$ as shown in Figure 4(a), which reserves the good effect of $\boldsymbol{d}$.
- **Action**: conduct $do(D = \boldsymbol{0})$ to estimate $\boldsymbol{e}_2'$ and $\boldsymbol{z}_1'$ as in Figure 4(b). $do(D = \boldsymbol{0})$ means an empty interaction history[5], and $\boldsymbol{z}_1'$ is thus free from the impact of out-of-date interactions $\boldsymbol{d}$.
- **Prediction**: use $\boldsymbol{z}_1'$ and $\boldsymbol{z}_2$ to compute the interaction probability $\boldsymbol{d}' = f_{\theta_3}(\boldsymbol{z}_1', \boldsymbol{z}_2)$ as illustrated Figure 4(c).

We summarize the detailed inference process in Algorithm 1.

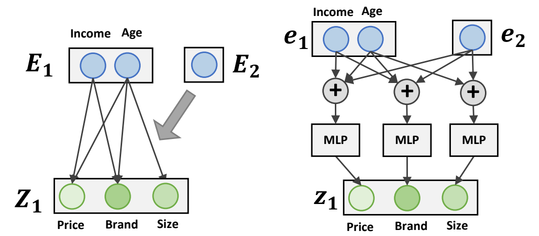
3.3 Fine-tuning for Fast Adaptation

We then consider the model adaptation once new user interactions are collected from the OOD environment. Our key belief to pursue fast adaptation is in reusing the unchanged user representations to the greatest extent. According to the causal graph in Figure 2, Z_2 will not be affected by the shifts of E_1. In this light, we reuse Z_2 and only fine-tune the VAE networks to update Z_1. Moreover, the functions in the VAE are built upon causal relationships. As indicated by previous research [32], these functions are inherently more stable under the intervention and require less data to adjust the deviation of the parameters from the IID to OOD environments [3, 32].

3.4 COR with Fine-grained Causal Graph

We additionally consider the situation that the fine-grained causal graph between E_1, E_2, and Z_1 is available, for example, *income* affects the user preference over *price* and *brand* rather than *size* as shown in Figure 5(a). Although the fine-grained causal graph is usually hard to acquire, it might be constructed by expert experience [25]. Undoubtedly, such causal relationships are beneficial for OOD generalization [32, 47]. In this light, we further extend COR to incorporate them into the decoder when estimating $p(\boldsymbol{z}_1|\boldsymbol{e}_1, \boldsymbol{e}_2)$. In particular, we replace the MLP $f_{\theta_1}(\cdot)$ with a *Neural Causal Models* (NCM) [47], which can encode the fine-grained causal relationships.

[5] Note that counterfactual inference is not equal to totally discarding user interactions because we use unchanged $\boldsymbol{z}_2$. Besides, $do(D = \boldsymbol{0})$ can be improved by keeping the most recent interactions, which is left for future exploration.



(a) Fine-grained causal graph. (b) Neural Causal Models.

Figure 5: Illustration of fine-grained causal graph and NCM.

We assume that a factor in z_1 aligns with the preference over an item feature (*e.g.,* price). As to one factor in z_1, NCM takes the sum of its parents' representations based on the causal graph. As shown in Figure 5(b), the representations of income, age, and $\boldsymbol{e}_2$ are summed for the preference over price. We then feed the sum to an MLP to predict Z_1. Note that we don't have explicit supervision over the alignment between z_1 and the preference over item features, which is learned implicitly based on the fine-grained causal graph during training. Besides, the MLP is shared across all factors in z_1 to reduce the number of parameters. NCM can be injected into the COR framework without altering other components.

4 EXPERIMENTS

In this section, we conduct extensive experiments to answer the following research questions:

- **RQ1:** How does COR perform in the OOD generalization task as compared to baselines?
- **RQ2:** How effective is the fine-tuning of COR on fast adaptation?
- **RQ3:** How do the different components (*e.g.,* counterfactual inference and fine-grained causal graph) affect the performance?

4.1 Experimental Settings

Datasets. We conduct experiments on one synthetic dataset and two real-world ones. As shown in Figure 6, we construct the synthetic dataset by following the user interaction generation process in the real world. Specifically,

1) User/item feature sampling: we assume 1,000 users and 1,000 items, where each user has an observed feature (*i.e., income*) and ten unobserved ones while each item has eight observed features (*i.e., type (2), brand (4), and price(2)*) and two unobserved ones. The unobserved user/item features are drawn from the standard Gaussian $\mathcal{N}(0, 1)$ [19] while user income is drawn from $\mathcal{N}(-1, 1)$. As to observed item features, type (*shoe* and *phone*), brand (*Nike* and *Apple*), and price (*high* and *low*) are discrete variables in $\{0, 1\}$ and sampled from Bernoulli distributions based on some causal relationships between item features. For example, *Apple* and *phone* easily lead to *high price.*
2) User preference estimation: after the sampling of user features, we assume the fine-grained causal relationships from user features to preference based on prior knowledge [25], such as a positive effect of income on the preference over high price. We have two types of relationships: positive and negative ones. They take the sum of user features by positive/negative weights to calculate user preference, where a sigmoid function is used after the sum to increase the non-linear complexity in the relationships.

Table 1: Statistics of the three datasets. Note that "int." denotes "interactions".

Dataset	#User	#Item	#IID int.	#OOD int.	Density
Synthetic data	1,000	1,000	145,270	112,371	0.257641
Meituan	2,145	7,189	11,400	6,944	0.001189
Yelp	7,975	74,722	305,128	99,525	0.000679

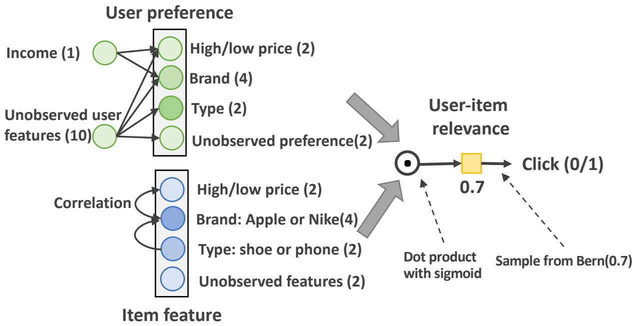


Figure 6: Illustration of constructing synthetic data.

Given user features, we then sample the ten-dimension user preference, *i.e.,* the preference over eight item features (8) and unknown preference (2).

3) User interaction sampling: once we obtain user preference $\boldsymbol{z}$ and item features $\boldsymbol{i}$, we can calculate user-item relevance r by $r = S(\boldsymbol{z}^T \boldsymbol{i})$ where $S(\cdot)$ is the *sigmoid* function. Next, we sample the interaction data from a Bernoulli distribution $Bern(r)$ [30].
4) OOD data collection: to collect user interactions in the OOD environment, we re-sample the income from $\mathcal{N}(1, 1)$ for each user, which significantly differs from the original income. Thereafter, we keep item features fixed and repeat the previous step 2 and 3 to sample new user interactions.

More details on the synthetic dataset are available in Appendix A.1. Besides, we also use two real-world datasets.

- Meituan is a public food recommendation dataset[6] with rich user/item features, such as user consumption level and food price. We consider the shifts of average consumption levels from weekdays to weekends as the drifted user feature, where many users have a higher/lower consumption in weekends than that of weekdays. Thus the user interactions (*i.e.,* purchases) in weekdays and weekends are regarded as IID and OOD interactions, respectively.
- Yelp is a popular restaurant recommendation dataset[7] where we treat the user location as the shifted feature. We first select the users with shifted locations (*e.g.,* state changes from Florida to Michigan). We then sort user ratings by timestamps, and divide them into two parts based on the changed state feature, where the two parts are used as IID and OOD interactions, respectively.

We only treat the interactions with ratings $\geq$ 4 as positive samples [38, 45]. For the IID or OOD interactions in each dataset, we separately split them into IID/OOD training, validation, and test sets by the ratio of 80%, 10%, and 10%. The fine-grained causal graph between E_1, E_2, and Z_1 is available in the synthetic data while it is unknown in the two real-world datasets. The statistics of the three datasets are shown in Table 1.

[6] https://www.biendata.xyz/competition/smp2021_2/.
[7] https://www.yelp.com/dataset.

Table 2: The comparison of OOD generalization performance between the baselines and COR on the three datasets. %improve. represents the relative improvement achieved by COR over the best results of the baselines. The best results are highlighted in bold while the second best ones are underlined.

Dataset	Synthetic Data					Meituan					Yelp				
IID/OOD tests	IID	OOD				IID	OOD				IID	OOD			
Metric	R@20	R@10	R@20	N@10	N@20	R@50	R@50	R@100	N@50	N@100	R@50	R@50	R@100	N@50	N@100
FM	0.3666	0.0572	0.1074	0.0604	0.0792	0.0846	0.0121	0.0205	0.0043	0.0057	0.1228	0.0964	0.1389	0.0313	0.0385
NFM	0.3629	0.0405	0.0761	0.0438	0.0560	0.0825	0.0233	0.0354	0.0066	0.0085	0.1222	0.0829	0.1276	0.0241	0.0316
MultiVAE	**0.3693**	0.0208	0.0408	0.0172	0.0257	0.1054	0.0238	0.0368	0.0069	0.0091	0.1399	0.0365	0.0582	0.0118	0.0154
MacridVAE	0.3573	0.0231	0.0392	0.0192	0.0262	0.1163	0.0219	0.0364	0.0067	0.0090	0.1526	0.0408	0.0634	0.0135	0.0174
MacridVAE+FM	0.3648	0.0463	0.0836	0.0513	0.0643	**0.1219**	0.0233	0.0364	0.0066	0.0087	0.1536	0.0407	0.0626	0.0140	0.0178
COR	0.3628	**0.0767**	**0.1443**	**0.0804**	**0.1056**	0.1159	**0.0368**	**0.0578**	**0.0101**	**0.0135**	**0.1539**	**0.1416**	**0.1986**	**0.0500**	**0.0595**
%Improve.	-0.57%	34.09%	34.36%	33.11%	33.33%	-4.92%	54.62%	57.07%	46.38%	48.35%	0.20%	46.89%	42.98%	59.74%	54.55%

Baselines. We compare COR with several competitive methods:

- **FM [27]** and **NFM [14]** are the most representative feature-based recommender models, which can use the shifted user features for OOD recommendation.
- **MultiVAE [19].** We adopt the VAE-based method, MultiVAE, which also considers the interaction generation procedure but ignores causal relationships $(E_1, E_2) \to (Z_1, Z_2) \to D$.
- **MacridVAE.** This is a representative method of disentangled recommendation, which learns the hierarchical user representations from historical interactions.
- **MacridVAE+FM.** Since MacridVAE neglects the user features, we enhance it by linearly combining the prediction scores of MacridVAE and FM in a late-fusion manner for re-ranking.

Evaluation. For a fair comparison, we tune the hyper-parameters of COR and baselines via the IID validation data in the task of OOD generalization because user interactions in OOD environments are unavailable; while for fast adaptation, OOD validation data is applied. The details on the hyper-parameter tuning are presented in Appendix A.3. To compare the performance, we adopt two popular metrics, Recall@K (R@K) and NDCG@K (N@K) [46]. They are used under the all-ranking protocol [41], which evaluates the top-K items selected from all items that are not interacted by the users. For the synthetic data, K is set as 10 and 20 while K is reported as 50 and 100 in Meituan and Yelp due to the large amount of items.

4.2 Overall Performance (RQ1 & RQ2)

4.2.1 *OOD Generalization*.

We train the baselines and COR on IID interactions and evaluate their performance on both IID and OOD tests, whose results on the three datasets are reported in Table 2. We omit more results on the IID tests with similar trends due to space limitation. From Table 2, we have the following observations:

- The performance drops sharply from the IID test to OOD test on the three datasets, which is due to the significant distribution shifts in OOD environments. Besides, the results of different methods are very close on the IID test while the variance is large under the OOD test. This indicates that the recommender models have comparable representation capability in IID environments, but the OOD generalization abilities are quite different.
- COR consistently yields superior performance on the three OOD tests with comparable or slightly better performance on the IID test. Table 2 shows that the relative improvements of COR over the best baseline on OOD settings are higher than 30% across the three datasets. The high OOD performance validates the strong generalization ability of COR under user feature shifts. This makes sense because COR resorts to causal modeling of the interaction generation procedure and estimates the post-intervention interaction probability. Besides, COR is unaffected by the out-of-date interactions due to counterfactual inference.
- FM and NFM outperform VAE-based methods *w.r.t.* OOD performance on the synthetic data and Yelp while having inferior OOD performance on Meituan. The reason is that the feature shifts are more significant on the synthetic data (*i.e.,* income) and Yelp (*i.e.,* location) while the changes of user consumption levels on Meituan are smaller. As such, feature-based baselines (*i.e.,* FM and NFM) have larger advantages on the datasets with notable feature shifts. In contrast, MultiVAE and MacridVAE perform better when there exist little feature shifts due to the superiority of modeling the interaction generation procedure [22].
- MacridVAE surpasses MultiVAE in most cases, especially on the OOD tests with large feature shifts, which is consistent with the claims in [22]. This observation justifies the rationality of learning disentangled representations: if a user feature changes, only partial representations need to be updated. In addition, the OOD performance of MacridVAE+FM is usually between MacridVAE and FM, illustrating that the simple fusion of multiple recommender models cannot effectively enhance the OOD generalization ability.

4.2.2 *Fast Adaptation*.

To evaluate the ability of fast adaptation, we assume that the OOD validation set and a small proportion of OOD training data are available during training. As such, we fine-tune the well-trained models by partial OOD training data and select the best model for OOD tests via OOD validation data. For COR, we reuse Z_2 and only optimize the VAE parameters to update Z_1. The results *w.r.t.* the interaction proportion varying from 0% to 30% are provided in Figure 7. The performance on Meituan is similar to that of Yelp, which is moved to Appendix A.2.

By comparing the performance, we observe the followings. 1) COR achieves better OOD performance by using less user interactions, especially on the sparse real-world datasets, which verifies the effectiveness of causal modeling and reusing Z_2 on fast adaptation. 2) The performance difference among different methods is gradually decreasing as the proportion increases, particularly on the dense synthetic data (*cf.* Table 1). This is rational because more user interactions will make the OOD environment become

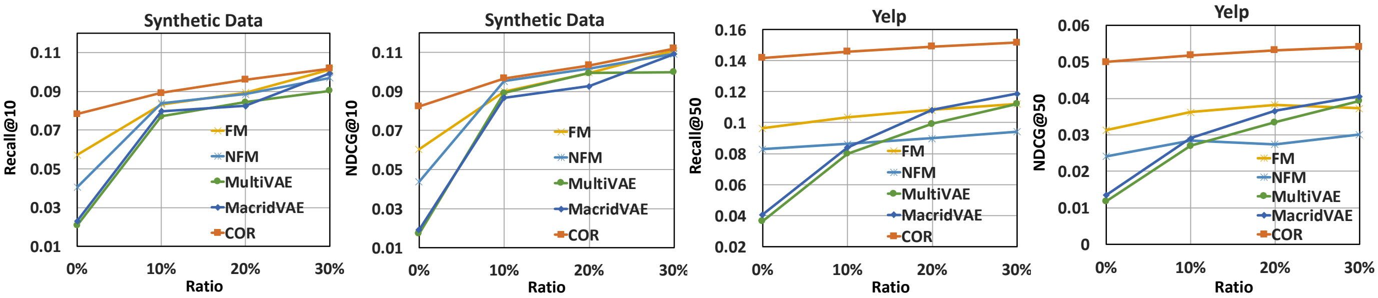


Figure 7: Fine-tuning performance of the baselines and COR *w.r.t.* different proportions of new user interactions collected from the OOD environment. We omit the performance of MacridVAE+FM because it is between the results of MacridVAE and FM.

Table 3: OOD performance of COR with (w/) and without (w/o) counterfactual inference on the three datasets.

OOD test	Variants	R@10	R@20	N@10	N@20
Synthetic Data	w/o CI	0.0401	0.0757	0.0416	0.0536
	w/ CI	**0.0767**	**0.1443**	**0.0804**	**0.1056**
OOD test	**Variants**	**R@50**	**R@100**	**N@50**	**N@100**
Meituan	w/o CI	0.0294	0.0545	0.0080	0.0121
	w/ CI	**0.0368**	**0.0578**	**0.0101**	**0.0135**
Yelp	w/o CI	0.1383	0.1960	0.0489	0.0585
	w/ CI	**0.1416**	**0.1986**	**0.0500**	**0.0595**

Table 4: Effect of the fine-grained causal graph (FGCG).

	Variants	R@10	R@20	N@10	N@20
IID test	w/o FGCG	0.2192	0.3494	0.3688	0.3649
	w/ FGCG	**0.2318**	**0.3628**	**0.3976**	**0.3856**
OOD test	w/o FGCG	0.0587	0.1114	0.0687	0.0874
	w/ FGCG	**0.0767**	**0.1443**	**0.0804**	**0.1056**

a new IID environment and the representation capacity of these methods in IID environments is comparable, which is consistent with the findings in Table 2. Besides, the quicker increase on the synthetic data is because this dataset is more dense and its interaction pattern is clearer. 3) MacridVAE and MultiVAE have the sharper performance rise than FM and NFM on Meituan and Yelp. This is attributed to that FM and NFM are user-based recommender models [14], and thus their embeddings are more affected by the out-of-date interactions.

4.3 In-depth Analysis (RQ3)

To further explore COR, we design ablation experiments, case studies, and the visualization of user and item representations.

4.3.1 ***Ablation Study.*** We conduct ablation studies to analyze the effects of counterfactual inference and fine-grained causal graph.

• **Ablation of Counterfactual Inference.** We disable the strategy of counterfactual inference by using $D = \boldsymbol{d}$ instead of $D = \boldsymbol{0}$ to estimate Z_1 during the inference stage. The results of COR with counterfactual inference (*i.e.,* w/ CI) and without it (*i.e.,* w/o CI) are summarized in Table 3. From the table, we can find that the OOD performance consistently decreases when counterfactual inference is disabled, which is possibly due to counterfactual inference blocking the effect of out-of-date interactions $\boldsymbol{d}$ on Z_1, and thus preventing COR from recommending inappropriate items.

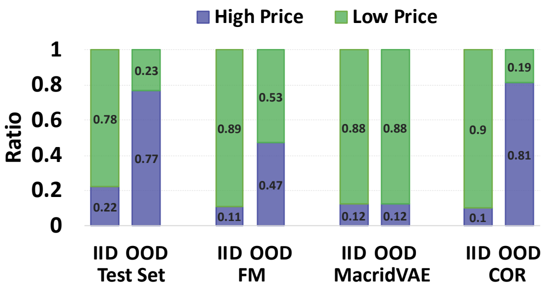


Figure 8: Visualization of the recommendations changed from IID to OOD environments.

• **Ablation of Fine-grained Causal Graph (FGCG).** We assume that the fine-grained causal graph between E_1, E_2, and Z_2 is unavailable on the synthetic dataset to analyze its effect. From the results in Table 4, we observe that: 1) the performance drops on both IID and OOD tests if FGCG is removed, which verifies its effectiveness. An explanation is that FGCG provides fine-grained effects of user features, leading to more accurate preference learning. 2) As compared to the IID test, the relative performance decrease is larger on the OOD test, partly validating the superiority of using causal relationships on OOD generalization. 3) The OOD performance without FGCG still surpasses the best baseline in Table 2, highlighting that the significant improvement on the synthetic dataset is not just attributed to FGCG. Instead of NCM, the MLP $f_{\theta_1}(\cdot)$ also shows promising OOD generalization ability.

4.3.2 ***Case Study.*** To illustrate how COR achieves notable OOD performance, we analyze the recommendation results of different methods for some users on the synthetic dataset. Specifically, we select 477 low-income users who have an income increase larger than a threshold (*i.e.,* 2) in the OOD environment. We then separately collect their positive items in the IID/OOD tests and the recommended top-20 items by representative FM, MacridVAE, and COR to visualize the item distribution over different prices.

From Figure 8, we find a dramatic preference increase over the high-price items from IID to OOD tests. As to the results of FM, MacridVAE, and COR, the observations are as follows. 1) All of them recommend a large proportion of low-price items to these low-income users in the IID environment, which explains the excellent performance on the IID test. 2) For OOD recommendation, MacridVAE has the same recommendations as the IID one because

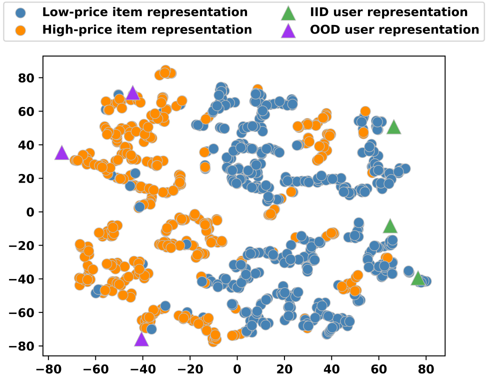


Figure 9: Visualization of user representations varying from IID to OOD environments. Best view in color.

it does not utilize the shifted user feature. FM recommends more high-price items by using user features but still suffers from the out-of-date interactions, causing many recommendations of low-price items. In contrast, the proportion of high-price items recommended by COR is the most similar to that of the OOD test. This highlights that COR not only captures the invariant causal relationships from income to the preference over price, but also mitigates the effect of out-of-date interactions.

For further analysis, we visualize the user/item representations of COR by t-SNE [34] in Figure 9. We take the weights in the last layer of $f_{\theta_3}(\cdot)$ as the item representations, and treat the vector obtained from z_1 and z_2 via a fully-connected layer as user representation. Three low-income users who have significant income increases in the OOD environment are randomly selected for visualization. From Figure 9, we find that: 1) high-price and low-price item representations are well disentangled, and 2) user representations move closer to high-price items in the OOD environment, which intuitively explains the superior OOD performance of COR.

5 RELATED WORK

Causal Recommendation. Data-driven recommender systems have achieved great success in alleviating the issue of information overload [8, 21]. However, they usually assume the IID assumption and amplify the bias in training data [5, 39], which might induce unfairness [1, 7], cause filter bubbles [12], and decrease the generalization ability in OOD environments [32]. So far, many researchers have tried to incorporate causality into deep learning [9, 10, 52]. As to causal recommender models [4, 58], they are mainly based on two causal frameworks: *potential outcome framework* [29] and *structural causal models* [25]. For the potential outcome framework, two most representative techniques are *inverse propensity scoring* [30] and *doubly robust* [42], which are widely used to debias explicit [31] and implicit feedback [54] for unbiased recommendation. As to structural causal models, existing work usually scrutinizes the causal relationships via causal graph and utilizes intervention [39, 55] or counterfactual inference [40, 44, 53, 58] to estimate causal effect [26] for debiasing, fairness, and explanation. Nevertheless, previous causal methods ignore the OOD recommendation issue of data-driven models, leading to poor generalization in the OOD environments.

Disentangled Recommendation. Existing work on disentangled recommendation learns factorized user/item representations to capture the complex factors behind user-item interactions [6, 22], where disentangled representations are more robust to distribution shifts. In particular, previous studies either utilize the VAE framework for disentanglement [22, 24, 35] or encourage the independence of multiple user representations in traditional recommender models [15, 23, 41, 43]. For example, MacridVAE [22] learns hierarchical representations from historical interactions: high-level user intention and low-level user preference. However, these studies neglect causality and heavily depend on the anti-causal modeling from D to (Z_1, Z_2) for representation learning, thus suffering from the issue of out-of-date interactions.

Model Adaptation in Recommendation. Distribution shifts widely exist in the recommendation scenarios. The popular solution is model re-training [28] while it needs sufficient new interactions and training time [56]. To tackle this, *model adaption* has been proposed to improve the adaptation ability by using less data in the tasks of cross-domain recommendation [57] and cold-start problem [50]. Technically, model adaption is usually implemented by parameter patch [33, 50], feature transformation [20], and meta learning [18, 49]. However, different from cross-domain recommendations, OOD recommendation in this work focuses on the items in a single domain with user feature shifts in OOD environments. Besides, only partial user features and preference are changed, which differs from the cold-start problem with new users/items. Therefore, how to improve the OOD generalization and fast adaption abilities of recommender models under user feature shifts is still untapped to date.

6 CONCLUSION AND FUTURE WORK

In this work, we formulated the OOD recommendation problem from a causal view, where user feature shifts are formulated as intervention and OOD recommendation aims to estimate the post-intervention interaction probability. Furthermore, we developed two objectives for OOD recommendation: strong OOD generalization and fast adaptation. To this end, we inspected the generation procedure from features to interactions and proposed a novel COR framework to perform causal modeling of the procedure. Based on the COR framework, we conducted post-intervention inference and utilized counterfactual inference to mitigate the effect of out-of-date interactions. Moreover, a fine-tuning strategy is applied for fast adaptation. Extensive experiments on three datasets validate the effectiveness and rationality of incorporating causal representation learning for OOD generalization and fast adaptation.

This work makes the initial attempt to explore OOD recommendation by causal representation learning, leaving many promising directions to future work. Specifically, 1) it is non-trivial to study the shifts of unobserved user features, and hence the performance of COR on unobserved shifts should be explored. 2) This work ignores the causal relationships among user features and the relationships among different user preference. How to uncover these fine-grained causal relationships for better recommendation is valuable. 3) An effective way of incorporating item features into COR is worth studying because they might help to capture user preference over item categories.

REFERENCES

[1] Himan Abdollahpouri, Masoud Mansoury, Robin Burke, and Bamshad Mobasher. 2020. The Connection Between Popularity Bias, Calibration, and Fairness in Recommendation. In *RecSys*. ACM, 726–731.

[2] Shoshana Abramovich and Lars-Erik Persson. 2016. Some New Estimates of the 'Jensen Gap'. *J Inequal Appl* 2016, 1 (2016), 1–9.

[3] Yoshua Bengio, Tristan Deleu, Nasim Rahaman, Nan Rosemary Ke, Sébastien Lachapelle, Olexa Bilaniuk, Anirudh Goyal, and Christopher J. Pal. 2020. A Meta-Transfer Objective for Learning to Disentangle Causal Mechanisms. In *ICLR*.

[4] Stephen Bonner and Flavian Vasile. 2018. Causal Embeddings for Recommendation. In *RecSys*. ACM, 104–112.

[5] Allison JB Chaney, Brandon M Stewart, and Barbara E Engelhardt. 2018. How Algorithmic Confounding in Recommendation Systems Increases Homogeneity and Decreases Utility. In *RecSys*. ACM, 224–232.

[6] Zeyu Cui, Feng Yu, Shu Wu, Qiang Liu, and Liang Wang. 2021. Disentangled Item Representation for Recommender Systems. *TIST* 12, 2 (2021), 1–20.

[7] Cyrus DiCiccio, Sriram Vasudevan, Kinjal Basu, Krishnaram Kenthapadi, and Deepak Agarwal. 2020. Evaluating Fairness Using Permutation Tests. In *KDD*. ACM, 1467–1477.

[8] Yujuan Ding, Yunshan Ma, Wai Keung Wong, and Tat-Seng Chua. 2021. Leveraging Two Types of Global Graph for Sequential Fashion Recommendation. In *ICMR*. ACM, 73–81.

[9] Fuli Feng, Weiran Huang, Xin Xin, Xiangnan He, and Tat-Seng Chua. 2021. Should Graph Convolution Trust Neighbors? A Simple Causal Inference Method. In *SIGIR*. ACM.

[10] Fuli Feng, Jizhi Zhang, Xiangnan He, Hanwang Zhang, and Tat-Seng Chua. 2021. Empowering Language Understanding with Counterfactual Reasoning. In *ACL-IJCNLP Findings*. ACL.

[11] Xiang Gao, Meera Sitharam, and Adrian E. Roitberg. 2019. Bounds on the Jensen Gap, and Implications for Mean-Concentrated Distributions. *AJMAA* 16, 14 (2019), 1–16. Issue 2.

[12] Yingqiang Ge, Shuya Zhao, Honglu Zhou, Changhua Pei, Fei Sun, Wenwu Ou, and Yongfeng Zhang. 2020. Understanding Echo Chambers in E-Commerce Recommender Systems. In *SIGIR*. ACM, 2261–2270.

[13] Samuel Gershman and Noah Goodman. 2014. Amortized Inference in Probabilistic Reasoning. In *CogSci*, Vol. 36.

[14] Xiangnan He and Tat-Seng Chua. 2017. Neural Factorization Machines for Sparse Predictive Analytics. In *SIGIR*. ACM, 355–364.

[15] Hao Jiang, Wenjie Wang, Yinwei Wei, Zan Gao, Yinglong Wang, and Liqiang Nie. 2020. What Aspect Do You Like: Multi-Scale Time-Aware User Interest Modeling for Micro-Video Recommendation. In *MM*. ACM, 3487–3495.

[16] Diederik P Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. In *arXiv:1412.6980*.

[17] Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *ICLR*.

[18] Hoyeop Lee, Jinbae Im, Seongwon Jang, Hyunsouk Cho, and Sehee Chung. 2019. Melu: Meta-learned User Preference Estimator for Cold-start Recommendation. In *KDD*. ACM, 1073–1082.

[19] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. 2018. Variational Autoencoders for Collaborative Filtering. In *WWW*. ACM, 689–698.

[20] Xixun Lin, Jia Wu, Chuan Zhou, Shirui Pan, Yanan Cao, and Bin Wang. 2021. Task-adaptive Neural Process for User Cold-Start Recommendation. In *WWW*. ACM, 1306–1316.

[21] Fan Liu, Zhiyong Cheng, Lei Zhu, Zan Gao, and Liqiang Nie. 2021. Interest-aware Message-Passing GCN for Recommendation. In *WWW*. ACM, 1296–1305.

[22] Jianxin Ma, Chang Zhou, Peng Cui, Hongxia Yang, and Wenwu Zhu. 2019. Learning Disentangled Representations for Recommendation. In *NeurIPS*. Curran Associates, Inc., 5712–5723.

[23] Jianxin Ma, Chang Zhou, Hongxia Yang, Peng Cui, Xin Wang, and Wenwu Zhu. 2020. Disentangled Self-supervision in Sequential Recommenders. In *KDD*. ACM, 483–491.

[24] Preksha Nema, Alexandros Karatzoglou, and Filip Radlinski. 2021. Disentangling Preference Representations for Recommendation Critiquing with ß-VAE. In *CIKM*. ACM, 1356–1365.

[25] Judea Pearl. 2009. *Causality*. Cambridge university press.

[26] Judea Pearl and Dana Mackenzie. 2018. *The Book of Why: The New Science of Cause and Effect* (1st ed.). Basic Books, Inc.

[27] Steffen Rendle. 2010. Factorization Machines. In *ICDM*. IEEE, 995–1000.

[28] Steffen Rendle and Lars Schmidt-Thieme. 2008. Online-updating Regularized Kernel Matrix Factorization Models for Large-scale Recommender Systems. In *RecSys*. ACM, 251–258.

[29] Donald B Rubin. 2005. Causal Inference Using Potential Outcomes: Design, Modeling, Decisions. *JASA* 100, 469 (2005), 322–331.

[30] Yuta Saito, Suguru Yaginuma, Yuta Nishino, Hayato Sakata, and Kazuhide Nakata. 2020. Unbiased Recommender Learning from Missing-Not-At-Random Implicit Feedback. In *WSDM*. ACM, 501–509.

[31] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations as Treatments: Debiasing Learning and Evaluation. In *ICML*, Vol. 48. JMLR.org, 1670–1679.

[32] Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. 2021. Toward Causal Representation Learning. *Proc. IEEE* 109, 5 (2021), 612–634.

[33] Xiang-Rong Sheng, Liqin Zhao, Guorui Zhou, Xinyao Ding, Binding Dai, Qiang Luo, Siran Yang, Jingshan Lv, Chi Zhang, Hongbo Deng, et al. 2021. One Model to Serve All: Star Topology Adaptive Recommender for Multi-Domain CTR Prediction. In *CIKM*. ACM, 4104–4113.

[34] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing Data Using t-SNE. *JMLR* 9, 11 (2008).

[35] Haonan Wang, Chang Zhou, Carl Yang, Hongxia Yang, and Jingrui He. 2021. Controllable Gradient Item Retrieval. In *WWW*. ACM, 768–777.

[36] Qinyong Wang, Hongzhi Yin, Zhiting Hu, Defu Lian, Hao Wang, and Zi Huang. 2018. Neural Memory Streaming Recommender Networks with Adversarial Training. In *KDD*. ACM, 2467–2475.

[37] Tan Wang, Jianqiang Huang, Hanwang Zhang, and Qianru Sun. 2020. Visual Commonsense r-cnn. In *CVPR*. IEEE, 10760–10770.

[38] Wenjie Wang, Fuli Feng, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. 2021. Denoising implicit feedback for recommendation. In *WSDM*. ACM, 373–381.

[39] Wenjie Wang, Fuli Feng, Xiangnan He, Xiang Wang, and Tat-Seng Chua. 2021. Deconfounded Recommendation for Alleviating Bias Amplification. In *KDD*. ACM, 1717–1725.

[40] Wenjie Wang, Fuli Feng, Xiangnan He, Hanwang Zhang, and Tat-Seng Chua. 2021. Click can be Cheating: Counterfactual Recommendation for Mitigating Clickbait Issue. In *SIGIR*. ACM, 1288–1297.

[41] Xiang Wang, Hongye Jin, An Zhang, Xiangnan He, Tong Xu, and Tat-Seng Chua. 2020. Disentangled Graph Collaborative Filtering. In *SIGIR*. ACM, 1001–1010.

[42] Xiaojie Wang, Rui Zhang, Yu Sun, and Jianzhong Qi. 2019. Doubly Robust Joint Learning for Recommendation on Data Missing Not at Random. In *ICML*. PMLR, 6638–6647.

[43] Yifan Wang, Suyao Tang, Yuntong Lei, Weiping Song, Sheng Wang, and Ming Zhang. 2020. DisenHAN: Disentangled Heterogeneous Graph Attention Network for Recommendation. In *CIKM*. ACM, 1605–1614.

[44] Zhenlei Wang, Jingsen Zhang, Hongteng Xu, Xu Chen, Yongfeng Zhang, Wayne Xin Zhao, and Ji-Rong Wen. 2021. Counterfactual Data-augmented Sequential Recommendation. In *SIGIR*. ACM, 347–356.

[45] Yinwei Wei, Xiang Wang, Liqiang Nie, Xiangnan He, and Tat-Seng Chua. 2020. Graph-Refined Convolutional Network for Multimedia Recommendation with Implicit Feedback. In *MM*. ACM, 3541–3549.

[46] Yinwei Wei, Xiang Wang, Liqiang Nie, Xiangnan He, Richang Hong, and Tat-Seng Chua. 2019. MMGCN: Multi-modal Graph Convolution Network for Personalized Recommendation of Micro-video. In *MM*. ACM, 1437–1445.

[47] Kevin Xia, Kai-Zhan Lee, Yoshua Bengio, and Elias Bareinboim. 2021. The Causal-Neural Connection: Expressiveness, Learnability, and Inference. In *NeurIPS*. Curran Associates, Inc.

[48] Mengyue Yang, Furui Liu, Zhitang Chen, Xinwei Shen, Jianye Hao, and Jun Wang. 2021. CausalVAE: Disentangled Representation Learning via Neural Structural Causal Models. In *CVPR*. IEEE, 9593–9602.

[49] Runsheng Yu, Yu Gong, Xu He, Yu Zhu, Qingwen Liu, Wenwu Ou, and Bo An. 2021. Personalized Adaptive Meta Learning for Cold-start User Preference Prediction. In *AAAI*. AAAI Press, 10772–10780.

[50] Fajie Yuan, Xiangnan He, Alexandros Karatzoglou, and Liguang Zhang. 2020. Parameter-efficient Transfer from Sequential Behaviors for User Modeling and Recommendation. In *SIGIR*. ACM, 1469–1478.

[51] Cheng Zhang, Kun Zhang, and Yingzhen Li. 2020. A Causal View on Robustness of Neural Networks. In *NeurIPS*, Vol. 33. Curran Associates, Inc.

[52] Shengyu Zhang, Tan Jiang, Tan Wang, Kun Kuang, Zhou Zhao, Jianke Zhu, Jin Yu, Hongxia Yang, and Fei Wu. 2020. DeVLBert: Learning Deconfounded Visio-Linguistic Representations. In *MM*. ACM, 4373–4382.

[53] Shengyu Zhang, Dong Yao, Zhou Zhao, Tat-Seng Chua, and Fei Wu. 2021. CauseRec: Counterfactual User Sequence Synthesis for Sequential Recommendation. In *SIGIR*. ACM, 367–377.

[54] Wenhao Zhang, Wentian Bao, Xiao-Yang Liu, Keping Yang, Quan Lin, Hong Wen, and Ramin Ramezani. 2020. Large-Scale Causal Approaches to Debiasing Post-Click Conversion Rate Estimation with Multi-Task Learning. In *WWW*. ACM, 2775–2781.

[55] Yang Zhang, Fuli Feng, Xiangnan He, Tianxin Wei, Chonggang Song, Guohui Ling, and Yongdong Zhang. 2021. Causal Intervention for Leveraging Popularity Bias in Recommendation. In *SIGIR*. ACM, 11–20.

[56] Yang Zhang, Fuli Feng, Chenxu Wang, Xiangnan He, Meng Wang, Yan Li, and Yongdong Zhang. 2020. How to Retrain Recommender System? A Sequential Meta-learning Method. In *SIGIR*. ACM, 1479–1488.

[57] Yongchun Zhu, Zhenwei Tang, Yudan Liu, Fuzhen Zhuang, Ruobing Xie, Xu Zhang, Leyu Lin, and Qing He. 2022. Personalized Transfer of User Preferences for Cross-domain Recommendation. In *WSDM*. ACM.

[58] Hao Zou, Peng Cui, Bo Li, Zheyan Shen, Jianxin Ma, Hongxia Yang, and Yue He. 2020. Counterfactual Prediction for Bundle Treatment. In *NeurIPS*.

A APPENDIX

A.1 Synthetic Data Construction

In this section, we detail the construction of the synthetic dataset, which covers four steps: 1) user/item feature sampling, 2) user preference estimation, 3) user interaction sampling, and 4) OOD data collection. Before explaining the construction algorithm, we introduce some concepts and notations. Specifically,

- U and I are the numbers of users and items, respectively.
- $\boldsymbol{e}_1 \in \mathcal{R}^{H_1}$ and $\boldsymbol{e}_2 \in \mathcal{R}^{H_2}$ denote the observed and unobserved features of user u, respectively.
- $\boldsymbol{i}_1 \in \mathcal{R}^{H_3}$ and $\boldsymbol{i}_2 \in \mathcal{R}^{H_4}$ denote the observed and unobserved features of item i, respectively.
- $z_1 \in \mathcal{R}^{H_3}$ and $z_2 \in \mathcal{R}^{H_4}$ represent the preference of user u, where $[z_1, z_2]$ aligns with $[\boldsymbol{i}_1, \boldsymbol{i}_2]$.
- $\boldsymbol{\lambda} \in \mathcal{R}^{H_3}$ denotes the parameters of the Bernoulli distribution for $\boldsymbol{i}_1$, which has some correlations defined by prior knowledge. For example, some brands are easy to produce high-price items.
- $\mu_1(\boldsymbol{e}_1, \boldsymbol{e}_2)$ is a structural function to calculate the mean of z_1 based on the causal relationships from $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ to z_1. Similarly, we have $\mu_2(\boldsymbol{e}_2)$ for z_2.

Thereafter, we present the construction algorithm in Algorithm 2. The implementation details on the hyper-parameters and structural functions can be found in the released code.

Algorithm 2 Synthetic Data Construction

```
Input: $U$, $I$; the dimension hyper-parameters $H_1$, $H_2$, $H_3$, and $H_4$;
    parameter $\boldsymbol{\lambda}$; structural functions $\mu_1(\cdot)$ and $\mu_2(\cdot)$.
 1: for each user $u$ in $\{1, 2, ..., U\}$ do                ▷ User feature sampling
 2:     draw $\boldsymbol{e}_1$ from $\mathcal{N}(-1, \mathbf{I}_{H_1})$;
 3:     draw $\boldsymbol{e}_2$ from $\mathcal{N}(0, \mathbf{I}_{H_2})$;
 4: end for
 5: for each item $i$ in $\{1, 2, ..., I\}$ do                ▷ Item feature sampling
 6:     draw $\boldsymbol{i}_1 \in \mathcal{R}^{H_3}$ from Bern($\boldsymbol{\lambda}$);
 7:     draw $\boldsymbol{i}_2$ from $\mathcal{N}(0, \mathbf{I}_{H_4})$;
 8: end for
 9: for each user $u$ in $\{1, 2, ..., U\}$ do ▷ User preference estimation
10:     draw $z_1 \in \mathcal{R}^{H_3}$ from $\mathcal{N}(\mu_1(\boldsymbol{e}_1, \boldsymbol{e}_2), \text{diag}\{0.05\})$;
11:     draw $z_2 \in \mathcal{R}^{H_4}$ from $\mathcal{N}(\mu_2(\boldsymbol{e}_2), \text{diag}\{0.05\})$;
12: end for
13: for each user $u$ in $\{1, 2, ..., U\}$ do  ▷ User interaction sampling
14:     obtain $\boldsymbol{z} = [z_1, z_2]$;
15:     for each item $i$ in $\{1, 2, ..., I\}$ do
16:         obtain $\boldsymbol{i} = [\boldsymbol{i}_1, \boldsymbol{i}_2]$;
17:         calculate $r = S(\boldsymbol{z}^T \boldsymbol{i})$;         ▷ $S(\cdot)$ is a sigmoid function
18:         sample the interaction (i.e., 0/1) from Bern($r$);
19:     end for
20: end for
21: collect the interaction matrix over all users and items $\boldsymbol{R}$.
22: for each user $u$ in $\{1, 2, ..., U\}$ do              ▷ OOD data collection
23:     sample $\boldsymbol{e}_1$ from $\mathcal{N}(1, \mathbf{I}_{H_1})$;
24: end for
25: keep $\boldsymbol{e}_2$, $\boldsymbol{i}_1$, and $\boldsymbol{i}_2$ fixed and repeat line 9-21 to collect the
    interaction matrix $\boldsymbol{R}'$.
Output: Interaction matrices $\boldsymbol{R}$ and $\boldsymbol{R}'$.
```

A.2 Fine-tuning on Meituan

We present the fine-tuning results of different methods on Meituan in Figure 10, from which we have the findings similar to Yelp. Furthermore, we see that: 1) VAE-based methods are more sensitive to OOD training data, especially regarding NDCG. This reflects that they can quickly adapt to the OOD environments by using less data. 2) FM shows inferior performance on Meituan, possibly because the causal relationships from user features to preference are more complex so that simple linear models cannot fit them well.

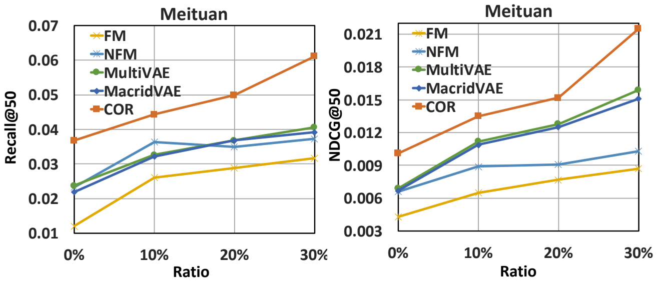


Figure 10: Fine-tuning performance on Meituan.

A.3 Hyper-Parameter Settings

Based on the default settings of baselines, we enlarge their hyper-parameter search scope and tune hyper-parameters as follows:

- **FM [27]/NFM [14]**. We tune the learning rate and normalization coefficient in $\{0.005, 0.01, 0.05\}$ and $\{0, 0.1, 0.2\}$, respectively. The hidden size is searched in $\{32, 64, 128, 512, 1024\}$. The negative sample number for each positive one is chosen from $\{1, 10, 30, 50\}$.
- **MultiVAE [19].** We follow the default settings, and additionally search the learning rate, weight decay, dropout ratio, hidden size, regularization parameter β in $\{0.0001, 0.001, 0.01\}$, $\{0, 0.01, 0.05\}$, $\{0.4, 0.5, 0.6\}$, $\{[200 \rightarrow 600 \rightarrow 200], [500 \rightarrow 800 \rightarrow 500], [1000 \rightarrow 1200 \rightarrow 1000], [2000 \rightarrow 5000 \rightarrow 2000]\}$, and $\{0.1, 0.2, ..., 1.0\}$, respectively.
- **MacridVAE.** In addition to the hyper-parameter searched in MultiVAE, we choose the number of macro factors and the scaling coefficient τ in $\{2, 10, 20\}$ and $\{0.05, 0.1, 0.2\}$, respectively.
- **MacridVAE+FM.** The linear hyper-parameter for fusion is tuned in $\{0.1, 0.2, ..., 0.9\}$.

As to the implementation of COR, we implement it by PyTorch and utilize Adam [16] with the early stopping strategy [55] for optimization. The learning rate is set as 0.001 and the batch size is 500. Besides, the weight decay, dropout ratio, KL coefficient β, and sample number L/M are searched in $\{0, 0.01, 0.05\}$, $\{0.4, 0.5, 0.6\}$, $\{0.1, 0.15, ..., 1.0\}$, and $\{1, 3, 5, 10\}$, respectively. Furthermore, the E_2 size is chosen from $\{20, 100, 400, 1000\}$ while the sizes of Z_1 and Z_2 are tuned in $\{10, 20, 100, 300\}$. The MLP network $g_\phi(\cdot)$ is tuned in $\{[1000], [2000]..., [5000], [1200, 1000]\}$. Besides, $f_{\theta_1}(\cdot)$, $f_{\theta_2}(\cdot)$, and $f_{\theta_3}(\cdot)$ are implemented by the fully-connected layer to save parameters, whose sizes are determined by the dimensions of E_1, E_2, Z_1, and Z_2. More details are available in the released code.